

## 2<sup>η</sup> Εργασία

### Ψηφιακή Επεξεργασία Εικόνας

#### Graph based-Image segmentation-Ncuts



Ασημακίδης Σταμάτιος 9711

Email: [asimakid@ece.auth.gr](mailto:asimakid@ece.auth.gr)

## 1ο ζητούμενο εργασίας

Στο συγκεκριμένο ερώτημα της εργασίας ζητείται η κατασκευή συνάρτησης που δέχεται σαν είσοδο μια εικόνα και επιστρέφει τον μη κατευθυντικό πίνακα affinity των pixel αυτής.

### Περιγραφή της συνάρτησης Image2Graph

Η εν λόγω συνάρτηση δημιουργήθηκε με στόχο να λειτουργεί τόσο για rgb όσο και για grayscale εικόνες. Συγκεκριμένα δημιουργείται αρχικά ένας μοναδιαίος πίνακας διαστάσεων  $(M*N)*(M*N)$  (καθώς τα στοιχεία της κυρίας διαγωνίου αντικατοπτρίζουν την ομοιότητα του pixel με τον εαυτό του) του οποίου το  $i,j$  στοιχείο εκφράζει την ομοιότητα του  $i$  pixel με το  $j$  pixel. Η αρίθμηση των pixel γίνεται διασχίζοντας την εικόνα κατα γραμμές, όπως δείχνεται και στον παρακάτω πίνακα.

$$\begin{pmatrix} 1 & \dots & cols \\ \vdots & \ddots & \vdots \\ (rows - 1) * cols + 1 & \dots & rows * cols \end{pmatrix}$$

Ενώ η συντεταγμένες του υπ αριθμό  $i$  pixel στην εικόνα προκύπτουν από τις παρακάτω εξισώσεις:

$$pixelrow = floor\left(\frac{i - 1}{cols}\right) + 1$$

$$pixelcol = mod(i - 1, cols) + 1$$

Έπειτα διασχίζοντας μέσω 2 for loops μόνο τα στοιχεία του πίνακα affinity πάνω από την κύρια διαγώνιο υπολογίζεται η ομοιότητα μεταξύ όλων των pixel της εικόνας. Η συγκεκριμένη δυνατότητα παρέχεται καθώς είναι γνωστό ότι ο πίνακας ομοιότητας των pixel είναι συμμετρικός, επομένως σε κάθε βήμα της εσωτερικής φορ συμπληρώνεται στοιχείο πάνω από την κύρια διαγώνιο όσο και το συμμετρικό (δηλ. Ομοιότητα $_{ij}$  = ομοιότητα $_{ji}$ ). Η μόνη διαφοροποίηση ανάμεσα στις rgb εικόνες και στις grayscale είναι ότι στην πρώτη περίπτωση η τιμή του κάθε στοιχείου του πίνακα υπολογίζεται με βάση την ευκλείδια απόσταση των διανυσμάτων της τριπλέτας rgb  $1/\exp(\text{ευκλείδια απόσταση})$ , ενώ για τις grayscale εικόνες αντι για την ευκλείδια απόσταση τίθεται στο εκθετικό η απόλυτη τιμή της διαφοράς της φωτεινότητας των δύο εικόνων.

## 2ο ζητούμενο εργασίας

Εδώ ζητείται η υλοποίηση της μεθόδου Graph spectral clustering και για τον σκοπό αυτό η δημιουργία συνάρτησης που λαμβάνει σαν είσοδο τον affinity πίνακα μιας εικόνας και τον αριθμό των επιθυμικών clusters.

## Περιγραφή της συνάρτησης `myGraphSpectralClustering`

Η παραπάνω υλοποιεί τον γνωστό αλγόριθμο όπως περιγράφεται στα πλαίσια της εκφώνησης. Αρχικά δημιουργείται ο διαγώνιος πίνακας που κάθε στοχείο της διαγωνίου εκφράζει το άθροισμα των στοχείων της κάθε γραμμής(θα μπορούσε και στηλης, δεν δημιουργεί κάποια διαφορά, λόγω συμμετρικού πίνακα), έπειτα ο πίνακας `L` ως διαφορά του `D` με τον πίνακα `affinity` όπως περιγράφεται στην εκφώνηση. Για τον υπολογισμό των γενικευμένων ιδιοδιανυσμάτων/ιδιοδιανυσμάτων χρησιμοποιείται η `eigs(L, D, k, 'smallestreal');`;

η όποια με τα συγκεκριμένα ορίσματα επιστρέφει τα `k` ιδιοδιανύσματα/ιδιοτιμές που αντιστοιχούν στις `k` μικρότερες ιδιοτιμές του πίνακας (οι οποίες γνωρίζουμε ότι είναι πραγματικές λόγω συμμετρίας των πινάκων). Τέλος καλείται η έτοιμη συνάρτηση του `matlab` `kmeans` από την οποία επιστρέφεται σε ποιο `cluster` ανήκει κάθε `πίξελ` της εικόνας (ένα διάνυσμα μήκους `NXM` που σε κάθε θέση του περιέχεται το `index` του `cluster` στο οποίο κατατάχθηκε το συγκεκριμένο `πίξελ`).

### Demo 1

Σε αυτό παρουσιάζονται τα αποτελέσματα της συνάρτησης που περιγράφηκε παραπάνω για ένα δεδομένο πίνακα που δίνεται μαζί με την εκφώνηση. Αρχικά για την σταθερότητα των αποτελεσμάτων (καθώς γενικά κάθε φορά ο `kmeans` δίνει διαφορετικά αποτελέσματα) δίνεται `rng(1)` όπως και στα υπόλοιπα `demo` που θα ακολουθήσουν.

Ακολουθούν τα αποτελέσματα της κλήσης της συνάρτησης για:

K=2	2	2	2	2	1	1	1	1	2	2	2	2
K=3	1	1	1	1	2	2	2	2	3	3	3	3
K=4	1	1	1	1	2	3	3	3	4	4	4	4

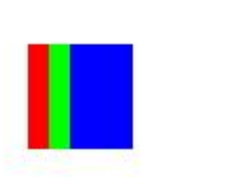
### Demo 2

Εδώ ζητείται η επείδξη των αποτελεσμάτων και της λειτουργικότητας και των δυο συναρτήσεων που έχουν περιγραφεί μέχρι στιγμής σε δύο δεδομένες εικόνες που ορίζονται από την εκφώνηση, εστί αρχικά δημιουργούνται οι γράφοι ομοιότητας για τις δύο εικόνες και χωρίζονται η κάθε μια σε 3 και 4 `segments`. Για λόγους οπτικοποίησης των αποτελεσμάτων των αποτελεσμάτων σε αυτό το σημείο δημιουργήθηκε μια επιπλέον συνάρτηση που αναπαρίστα με διαφορετικά χρώματα τα διαφορετικά `segments` της κάθε εικόνας. Η τελευταία ονόμαστηκε

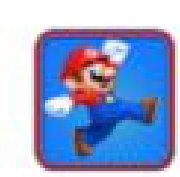
`showResultsnonRecursive` και δέχεται σαν όρισμα το διάνυσμα που επιστρέφει η `myspectralclustering`, τον αριθμό των κλάστερς που δημιουργήθηκαν και τις διαστάσεις της εικόνας και οπτικοποιεί τα αποτελέσματα.

Αρχικά παρατίθενται οι εικόνες εισόδου, ενώ ο πίνακας ομοιότητας που προκύπτει δεν παρουσιάζεται στην αναφορά της εργασίας λόγω των μεγάλων του διαστάσεων (2500x2500)

Εικόνα 1

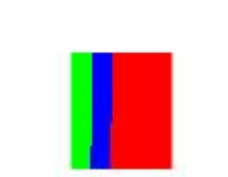


Εικόνα 2

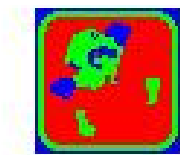


Εικόνα 1( 3 κλάστερ)

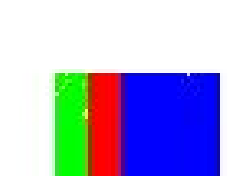
(δεν έχουν σχέση τα χρώματα που Αντιστοιχίζονται στα κλάστερ με τα χρώματα της εικόνας, γίνεται μόνο για λόγους οπτικοποίησης των αποτελεσμάτων με διαφορετικά χρώματα)



Εικόνα 2( 3 κλάστερ)



Εικόνα 1( 4 κλάστερ)



Εικόνα 2( 4 κλάστερ)



## Σχολιασμός των παραπάνω αποτελεσμάτων

Αρχικά φαίνεται η ορθότητα των αποτελεσμάτων για την εικόνα 1 για τα 3 κλάστερ που στην πραγματικότητα η εικόνα αποτελείται από 3 περιοχές που διαχωρίζονται με σωστό τρόπο. Όταν ωστόσο ζητείται από την συνάρτηση να χωριστεί σε περισσότερα τμήματα (που για την συγκεκριμένη εικόνα δεν είναι σωστό, λόγω της ύπαρξης 3 περιοχών στην εικόνα) ο αλγόριθμος διαχωρίζει ένα πολύ μικρό πλήθος πίξελ από την εικόνα τα οποία κατατάσσονται σε ένα τέταρτο κλάστερ, τα οποία στα παραπάνω αποτελέσματα παρουσιάζονται με κίτρινο χρώμα το οποίο σχεδόν δεν διακρίνεται στην εικόνα, λόγω του πολύ μικρού αριθμού πίξελ που ανήκουν σε αυτό, μιας και στην πραγματικότητα η ύπαρξη ενός τέταρτου κλάστερ δεν έχει νόημα. Για την δεύτερη εικόνα επιδεικνύεται η σωστή ταξινόμηση των πίξελ με βάση το χρώμα τους στην εικόνα, ενώ στην συγκεκριμένη περίπτωση η ύπαρξη ενός τέταρτου κλάστερ έχει μια φυσική σημασία καθώς επιτρέπει τον διαχωρισμό του «σώματος» του μάριο από το background της εικόνας, περιοχές που για 3 κλάστερ ανήκαν στο ίδιο κλάστερ, λόγω του ότι ήταν κοντά χρωματικά.

## 3ο ζητούμενο εργασίας

Στο συγκεκριμένο σημείο της εργασίας εισάγεται η έννοια της μετρηκής N-cut, η οποία θα χρησιμοποιηθεί ως κριτήριο τερματισμού της αναδρομικής διαδικασίας segmentation, το υπολογίζει την «συνδεσιμότητα/ομοιότητα» μεταξύ δύο νέων segments που δημιουργούνται.

## Περιγραφή της συνάρτησης `calculateNcut`

Στην οποία δίνονται σαν όρισμα ένας πίνακας affinity και το indexes των clusters στα οποία χωρίζονται τα στοιχεία του πίνακα. Αρχικά υπολογίζονται οι θέσεις των στοιχείων που ανήκουν στο πρώτο cluster και στο δεύτερο αντίστοιχα

```
indexesA = find(clusterIdx==1);
```

Εφόσον γνωρίζουμε ότι κάθε γραμμή (έστω η  $i$  γραμμή) του πίνακα αντιστοιχεί στις τιμές με τις οποίες συνδέεται το  $i$  στοιχείο με τα υπόλοιπα του γράφου μπορούμε να χωρίσουμε τον affinity σε δυο μέρη εκ των οποίων το ένα αντιστοιχεί στα στοιχεία του πρώτου κλαστερ και το άλλο του δεύτερου.

```
rowsforA = anAffinityMat(indexesA, :);
```

Το assocAV όπως ορίζεται στην εκφώνηση αντιστοιχεί άμεσα στο άθροισμα όλων των στοιχείων των πινάκων που προέκυψαν.

Για να κρατήσουμε τα βάρη με τα οποία συνδέονται μόνο μεταξύ τους τα στοιχεία του A και του B αντίστοιχα αρκεί από του παραπάνω πίνακες να κρατήσουμε μόνο τις στήλες που αντιστοιχούν στα στοιχεία του κλάστερ κάθε φορά.

```
matrixforallinA = rowsforA(:,indexesA);
```

Στην πραγματικότητα, ο τελευταίος αυτός πίνακας είναι ο affinity matrix των στοιχείων του segment A, και το άθροισμα όλων των στοιχείων του δίνει το assocAA. Πλέον μπορεί να υπολογιστεί το ncutValue το οποίο και επιστρέφεται από την συνάρτηση.

### Demo 3

3a) Για τις εικόνες εισόδου της εκφώνησης δημιουργούνται οι αντίστοιχοι πίνακες και χωρίζονται σε δύο τμήματα. Στην συνέχεια παρατίθενται οι τιμές ncut που προέκυψαν.

Για την πρώτη εικόνα

ncuta = 0.5092



Για την δεύτερη εικόνα

ncutb = 0.7853



Η τιμή της μετρικής αυτής για την πρώτη εικόνα προέκυψε μικρότερη καθώς ο διαχωρισμός των δύο segments έχει άμεση ουσία (η εικόνα είναι χωρισμένη ήδη σε 3 διαφορετικές χρωματικές περιοχές), οι οποίες δεν συσχετίζονται ιδιαίτερα μεταξύ τους. Στην δεύτερη εικόνα ο διαχωρισμός δεν είναι τόσο ευδιάκριτος (τα διαφορετικά τμήματα που δημιουργούνται είναι περισσότερο συσχετισμένα μεταξύ τους). Επομένως επιβεβαιώνεται αυτό που αναμένονταν ότι μεγαλύτερη τιμή του ncut συνεπάγεται και μεγαλύτερη συσχέτιση μεταξύ των τμημάτων της εικόνας που προέκυψαν. Αυτός είναι και ο λόγος για τον οποίον τίθεται ένα άνω φράγμα στην μετρική ncut πέρα από το δεν έχει νόημα ο περαιτέρω διαχωρισμός της εικόνας.

3b) Στο σημείο αυτό ζητείται η κατασκευή της αναδρομικής συνάρτησης που υλοποιεί τον αναδρομικό αλγόριθμο `ncuts`.

Για τον σκοπό αυτό δημιουργήθηκε η συνάρτηση `recursiveNcuts` η οποία τον υλοποιεί και επιστρέφει τα αποτελέσματα της σε μορφή πινάκων, η οποία προσπαθεί να προσγγίσει την δομή του δέντρου, μιας και μια τέτοια δομή δεν διατίθεται από το `matlab`.

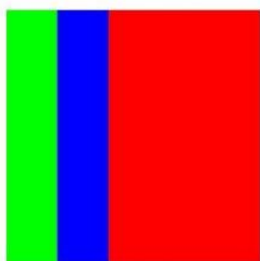
### Περιγραφή της συνάρτησης `recursiveNcuts`

Αρχικά αναφέρεται ότι η εν λόγω συνάρτηση δέχεται σαν όρια : έναν πίνακα `affinity`, το επίπεδο στο οποίο θα βρισκόμασταν στο δυαδικό δέντρο, τον δείκτη (αριθμό που αντιστοιχεί στην θέση) του πατέρα στο δυαδικό δέντρο (`from`), τα `threshold` για την λήξη της αναδρομικής διαδικασίας, καθώς και του πίνακες στους οποίους θα αποθηκευτούν τα αποτελέσματα. Οι τελευταίοι δίνονται σαν όρια για την υλοποίηση με αναδρομικό τρόπο (εναλλάκτικα θα μπορούσαν να οριστούν και σαν `global` μεταβλητές). Σαν πρώτο βήμα καλείται η `myGraphSpectralClustering` η οποία χωρίζει σε δυο `segment` τον πίνακα και υπολογίζεται η μετρική `ncut` για αυτά. Αν ικανοποιούνται οι συνθήκες για το όριο στην μετρική `ncut` και για το πλήθος των στοιχείων σε κάθε `segment`, τότε αποθηκεύονται τα αποτελέσματα με κατάλληλο τρόπο στους πίνακες και καλείται ξανά αναδρομικά η ίδια συνάρτηση μία φορά για κάθε `segment` που προέκυψε. Ο τρόπος αποθήκευσης των αποτελεσμάτων δεν παρουσιάζεται αναλυτικά στα πλαίσια αυτής της αναφοράς (για λόγους συντομίας μιας και δεν αφορά ιδιαίτερα τον σκοπό της εργασίας), αλλά αναφέρεται συνοπτικά η τελική δομή των αποτελεσμάτων. Ο πίνακας `array` περιέχει τα `indexes` των `clusters` που δημιουργούνται κάθε φορά, ο πίνακας `iarray` την θέση που θα είχε κάθε κλάστερ σε μια αναπαράσταση σε δυαδικό δέντρο και ο πίνακας `leaf` την δυαδική πληροφορία για το αν το κλάστερ δεν διαχωρίζεται επιπλέον δηλαδή αν στο δυαδικό δέντρο θα ήταν φύλλο. Στην πραγματικότητα τα τελικά αποτελέσματα της συνάρτησης αφορούν μόνο τα κλάστερ που βρίσκονται σε θέσεις φύλλα.

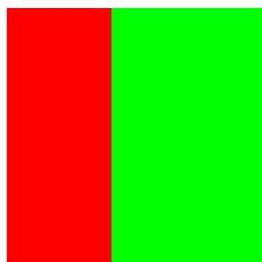
Για λόγους οπτικοποίησης των αποτελεσμάτων δημιουργήθηκε και εδώ μια συνάρτηση η οποία εκμεταλλεύεται την παραπάνω δομή αποτελεσμάτων και αναπαριστά τα διαφορετικά κλάστερ με διαφορετικά χρώματα, η λειτουργία της οποίας δεν θα περιγραφεί αναλυτικά για λόγους συντομίας της αναφοράς.

Αποτελέσματα για την πρώτη εικόνα

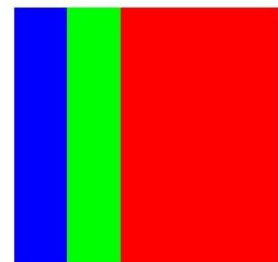
Αναδρομική με  $T1 = 5$   $T2 = 0.6$



Μη αναδρομική (2 seg)



Μη αναδρομική (3 seg)

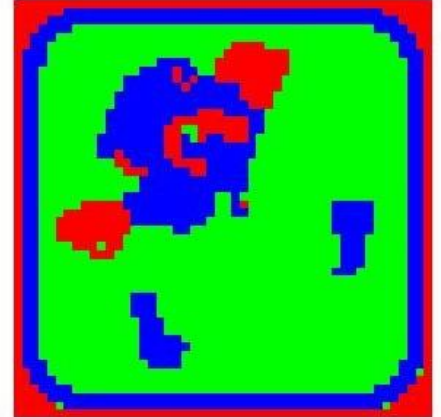
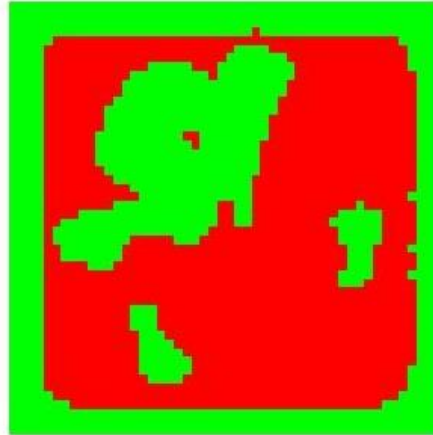
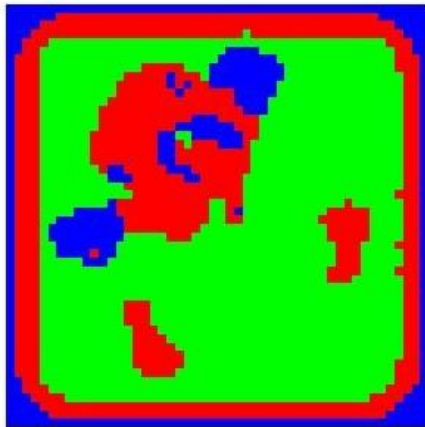


Αποτελέσματα για την δεύτερη εικόνα

Αναδρομική με  $T1 = 5$   $T2=0.8$

Μη αναδρομική(2 seg)

Μη αναδρομική(3 seg)

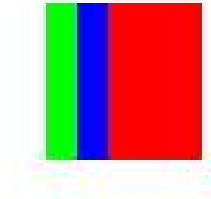


Στα παραπάνω αποτελέσματα στην περίπτωση της πρώτης εικόνας διακρίνεται η πλήρης ομοιότητα των αποτελεσμάτων της μη αναδρομικής μεθόδου με την αναδρομική με τον ίδιο αριθμό τμημάτων. Αυτό γίνεται γιατί η εικόνα αποτελεί την ιδανική περίπτωση εικόνας εισόδου προς τμηματοποίηση. Στην δεύτερη εικόνα μεταξύ της μη αναδρομικής για 3 segments παρατηρείται ομοιότητα με την αναδρομική, χωρίς ωστόσο τα αποτελέσματα να είναι ολόδια. Προκύπτει δηλαδή ότι η αναδρομική βασίζεται στα προηγούμενα αποτελέσματα στα επόμενα βήματα της, δηλαδή σπάει περαιτέρω τα υπάρχοντα τμήματα, όταν πληρούνται δεδομένες συνθήκες, όπως φαίνεται πχ στα πόδια του μαριο και στο περίγραμμα της εικόνας.

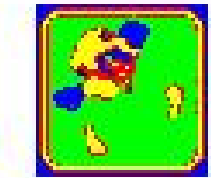
Ακολουθούν ορισμένα αποτελέσματα για διαφορετικές επιλογές κατωφλίων για λόγους παρουσίασης της επίδρασης αυτών στην αναδρομική διαδικασία.



Εικόνα 1 για  $T_2 = 1$ , παρα την μεγάλη αύξηση του  $\text{thres}$ , λόγω του ξεκάθολου διαχωρισμού τα αποτελέσματα παραμένουν ίδια.



Εικόνα 2 για  $T_2 = 0.9$ , εδώ το πλήθος των τμημάτων αυξήθηκε κατά 1. Επόμενως μπορούμε να υποθέσουμε/συμπεράνουμε ότι σε πραγματικές εικόνες εισόδου η διαδικασία είναι περισσότερο ευαίσθητη στην επιλογή των κατάλληλων  $\text{threshold}$ .



#### 4ο ζητούμενο εργασίας

Στο τελευταίο αυτό κομμάτι της εργασίας ζητείται εφαρμογή της παραπάνω μεθόδου σε μια πραγματική εικόνα εισόδου (αρκετά μεγαλύτερων διαστάσεων σε σχέση με τις προηφούμενες). Για να καταστεί εφικτός ο χειρισμός της εικόνας αυτή χωρίζεται αρχικά σε *superpixels*, και η μέθοδος τμηματοποίησης εφαρμόζεται εφαρμόζεται σε αυτά, διαφορετικά θα ήταν αδύνατος ο υπολογισμός λόγω των τεραστίων διαστάσεων του πίνακα *affinity*. Για τον χωρισμό σε *superpixels* χρησιμοποιείται έτοιμη συνάρτηση δημιουργίας αυτών που επιστρέφει έναν πίνακα διαστάσεων ίσων με της εικόνας σε κάθε του οποίου περιέχεται η τιμή το *label* του *superpixel* στο οποίο ανήκει το *πίξελ*. Για την αντιστοιχία της πληροφορίας σε μορφή χρώματος θα χρησιμοποιηθεί ο κανόνας της μέσης τιμής των *πίξελ* που ανήκουν στο κάθε *superpixel*, διαδικασία η οποία υλοποιείται από την συνάρτηση *superpixelDescriptor*.

## Περιγραφή της συνάρτησης `superpixelDescriptor`

Αυτή δέχεται σαν όρισμα την εικόνα και τον πίνακα με τα labels των superpixels στα οποία αντιστοιχεί κάθε πίξελ και επιστρέφει μια εικόνας με την αντιστοιχη τιμή χρώματος σε κάθε πίξελ με βάση το superpixel όπως ορίζεται από την εκφώνηση καθώς και μια λίστα με την τιμή του χρώματος που προέκυψε για τα superpixels. Το δεύτερο προστέθηκε παρόλο που δεν ορίζεται στην εκφώνηση καθώς η εν λόγω λίστα απαιτείται για την κατασκευή του affinity πίνακα των superpixels. Έτσι αρχικά στην συνάρτηση υπολογίζεται ο αριθμός των superpixel που δόθηκαν και μέσω μιας for διατρέχονται αυτά. Για κάθε ένα από αυτά εντοπίζονται οι συντεταγμένες των πίξελ στην εικόνα

```
[trows, tcols] = find(labels == i-1);
```

και στην μεταβλήτη tempsum κρατιέται το άθροισμα των χρωμάτων των πίξελ, για να προκύψει ο μέσος όρος σαν περιγραφέας χρώματος σε κάθε σουπερπίξελ. Έτσι η τιμή αυτή αποδίδεται σε όλες τις αντίστοιχες θέσεις στην εικόνα εξόδου και κάθε φορά ανανεώνεται η λίστα.

Έτσι η μπορεί να προκύψει μια αναπαράσταση της εικόνας εισόδου σε σουπερπίξελ, η οποία παρουσιάζεται παρακάτω.



Στην συνέχεια χρειάστηκε να τροποποιηθεί η συνάρτηση δημιουργίας του affinity matrix, ώστε να είναι δυνατή η λειτουργία της για λίστα απο superpixels. Έτσι δημιουργήθηκε η συνάρτηση `im2graphforlist`. Η συγκεκριμένη λειτουργεί με τον ίδιο τρόπο με την αντίστοιχη εκδοχή

της, όπως περιγράφηκε παραπάνω, με την διαφοροποίηση ότι η εικόνα/λίστα εισόδου υφίσταται κανονικοποίηση το διάστημα [0 1] (δηλαδή αναπαρίσταται σαν double εικόνα), όπως αυτό απαιτείται από το κριτήριο  $\exp(\text{distance})$ .

Έχοντας πλέον δημιουργήσει τον affinity πίνακα μπορούμε να χρησιμοποιήσουμε τις μεθόδους που δημιουργήθηκαν στα προηγούμενα ερωτήματα. Για την οπτικοποίηση των παραπάνω (εφόσον πλέον τα στοιχεία των πινάκων αφορούν superpixels) δημιουργήκαν οι συναρτήσεις

```
showResultsSupexels, showResultsSupexelsRecursive
```

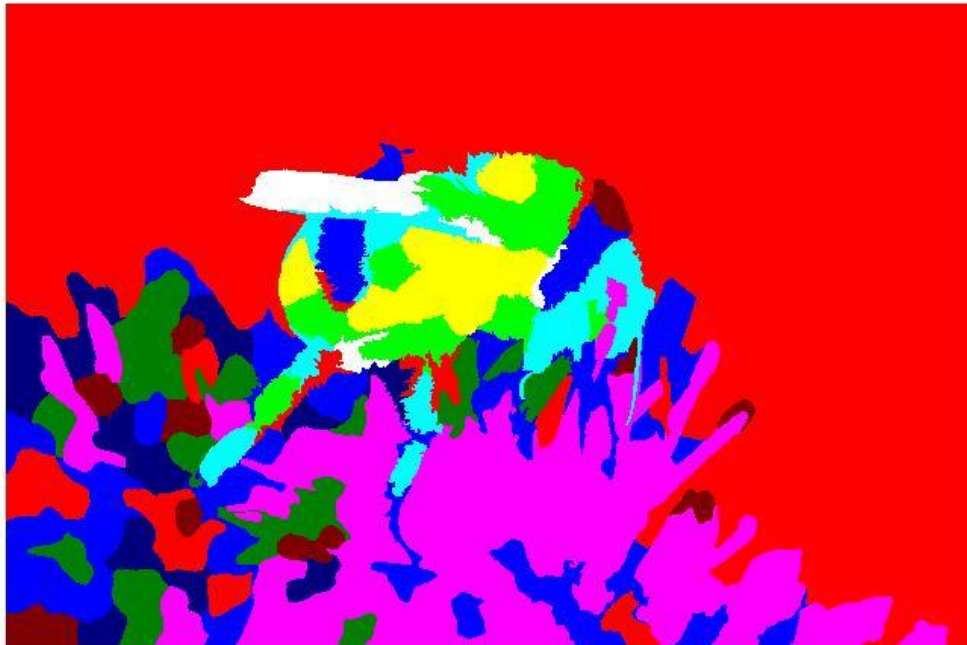
Οι οποίες αποτελούν κατάλληλες τροποποιήσεις προηγούμενων συναρτήσεων για αναπαράσταση των τμημάτων.

Παρουσίαση αποτελεσμάτων

Αναδρομική μέθοδος με  $T1 = 5$   $T2 = 0.95$  5 clusters



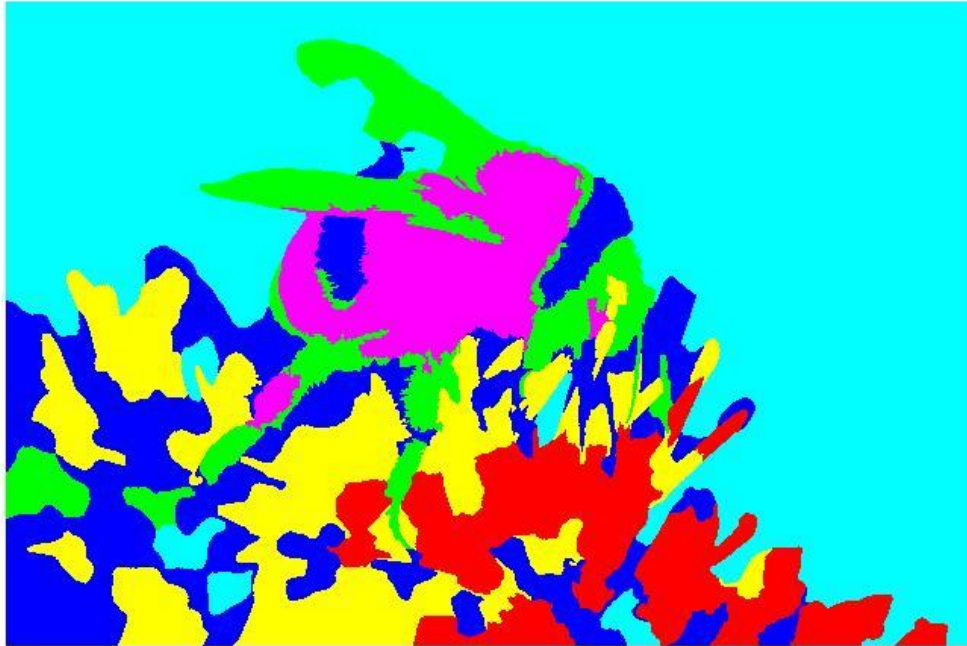
Αναδρομική μέθοδος με  $T1 = 5$   $T2 = 0.96$  10 clusters



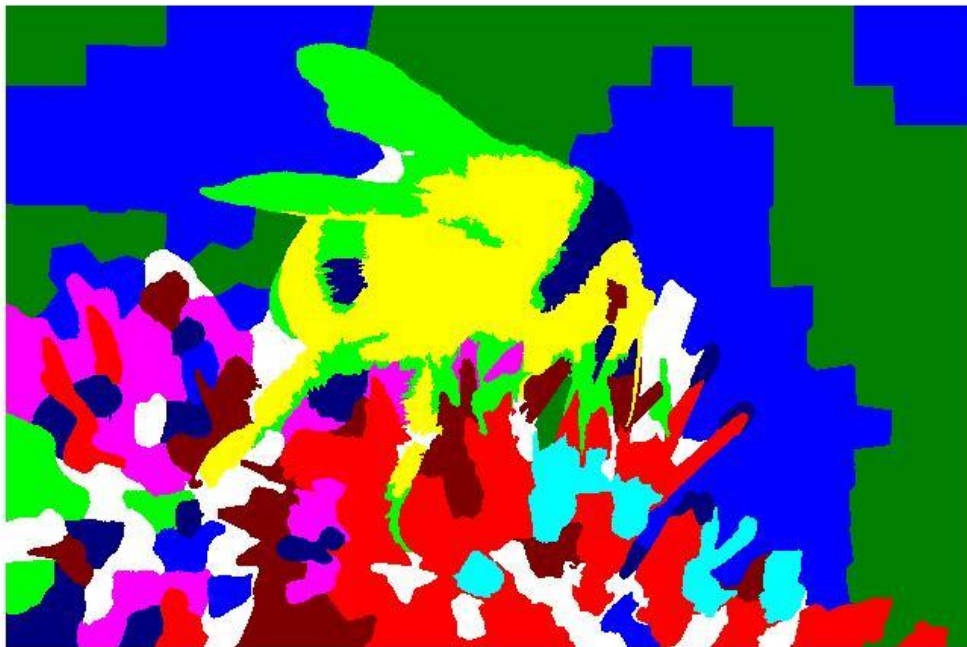
Αναδρομική μέθοδος με  $T1 = 5$   $T2 = 0.955$  7 clusters



Μη Αναδρομική μεθοδος με 6 clusters



Μη αναδρομική 10 clusters



Από τα παραπάνω φαίνεται ότι η αναδρομική διαδικασία είναι αρκετά πιο ευαίσθητη όσον αφορά την επιλογή των κατωφλίων για μεταβολή κατά 0.1 το πληθος των segments διπλασιάστηκε. Αυτό συμβαίνει άλλωστε γιατί το αν η διαδικασία δώσει σε ένα βήμα της κάποια συγκεκριμένη τιμή  $ncut$  δεν συνεπάγεται ότι σε επόμενα βήματα θα προκύπτουν όλο και μεγαλύτερες τιμές. Αύτος είναι και ο λόγος για τον οποίον προκύπτουν τα παραπάνω αποτελέσματα. Τόσο στις αναδρομικές εκδοχές παρουσιάζουν κάποια ομοιότητα και κάποιες διαφορές που οφείλονται στο γεγονός ότι η αναδρομική διαχωρίζει μόνο τα υπάρχοντα τμήματα.